CLAIMS

1. A method for executing a load locked and a store conditional instruction in a processor, comprising:

executing the load locked instruction to read a memory block, and said processor in response to executing the load locked instruction issuing a read modify system command to read said block and to take ownership of said block by said processor, and also setting a lock flag for the address of said memory block, and writing a value of said memory block into a cache of said processor as a cache copy of said memory block;

resetting said lock flag upon receipt of an invalidate message for said cache copy of said memory block, if any said invalidate messages are received by said processor;

waiting for a selected time interval before said processor surrenders ownership of said memory block upon receipt of an ownership request message, if any is received by said processor after execution of said load locked instruction;

executing the store conditional instruction, and said processor in response to executing said store conditional instruction testing said lock flag, and if said lock flag is set, writing to said cache copy of said memory block.

2. The method as in claim 1 further comprising:

checking the cache before executing the load locked instruction, and in the event that there is a cache miss, then proceeding to execute said load locked instruction, and in the event that there is a cache hit then reading the value from its cache and setting said lock flag.

3. The method as in claim 1 further comprising:
said invalidate messages are Inval Probes.

4. The method as in claim 1 further comprising:

said invalidate messages are forwarded RdMod probes.

5. The method as in claim 1 wherein said executing said store conditional instruction further comprises:

ending, in the event that said lock flag is reset, said store conditional instruction and not writing to said cache copy of said memory block.

6. The method as in claim 1 wherein said selected time interval further comprises:

a first time required for a predetermined number of cycles of a clock in said processor to retire.

7. The method as in claim 1 wherein said selected time interval further comprises:

a second time required for a predetermined number of assembly language instructions executing in said processor to retire.

8. The method as in claim 1 wherein said selected time interval further comprises:

a third time required until a store conditional instruction referring to said memory block executes.

9. The method as in claim 1 wherein said selected time interval further comprises:

a fourth time for all MAF entries in said processor MAF file to retire.

10. The method as in claim 1 wherein said selected time interval further comprises:

a fifth time during which a MAF entry for the LD_L instruction persists in a MAF file of said processor.

11. A computer executing a load locked and store conditional instructions, comprising:

a processor executing the load locked instruction to read a memory block, and said processor in response to executing the load locked instruction issuing a read modify system command to read said block and to take ownership of said block by said processor, and also setting a lock flag for the address of said memory block, and writing a value of said memory block into a cache of said processor as a cache copy of said memory block;

a MAF state machine resetting said lock flag upon receipt of an invalidate message for said cache copy of said memory block, if any said invalidate messages are received by said processor;

a controller waiting for a selected time interval before said processor surrenders ownership of said memory block upon receipt of an ownership request message, if any is received by said processor after execution of said load locked instruction;

said processor executing the store conditional instruction, and said processor testing said lock flag, and if said lock flag is set, writing to said cache copy of said memory block.

12. The apparatus as in claim 11 further comprising:

said processor checking the cache before executing the load locked instruction, and in the event that there is a cache miss then proceeding to execute said load locked instruction, and in the event that there is a cache hit then reading the value from its cache and setting said lock flag.

13. The apparatus as in claim 11 further comprising: said invalidate message is an Inval Probe.

14. The apparatus as in claim 11 further comprising: said invalidate message is a RdMod probe.

15. The apparatus as in claim 11 further comprising:

said processor ending, in the event that said lock flag is reset, said store conditional instruction and not writing to said cache copy of said memory block.

16. The apparatus as in claim 11 further comprising:

a first time required for a predetermined number of cycles of a clock in said processor to retire.

17. The apparatus as in claim 11 further comprising:

a second time required for a predetermined number of assembly language instructions executing in said processor to retire.

18. The apparatus as in claim 11 further comprising:

a third time required until a store conditional instruction referring to said memory block executes.

19. The apparatus as in claim 11 further comprising:

a fourth time for all MAF entries in said processor MAF file to retire.

20. The apparatus as in claim 11 further comprising:

a fifth time during which a MAF entry for the LD_L instruction persists in a MAF file of said processor.

21. A method of operating a multiprocessor computer system, comprising:

executing a load locked and a store conditional instruction in each processor in accordance with the following steps;

executing said load locked instruction to read a memory block, and said processor in response to executing the load locked instruction issuing a read modify system command to read said block and to take ownership of said block by said processor, and also setting a lock flag for the address of said memory block, and writing a value of said memory block into a cache of said processor as a cache copy of said memory block;

resetting said lock flag upon receipt of an invalidate message for said cache copy of said memory block, if any said invalidate messages are received by said processor;

waiting for a selected time interval before said processor surrenders ownership of said memory block upon receipt of an ownership request message, if any is received by said processor after execution of said load locked instruction;

executing the store conditional instruction, and said processor in response to executing said store conditional instruction testing said lock flag, and if said lock flag is set, writing to said cache copy of said memory block.

22. A multiprocessor computer system comprising:

a plurality of processor systems, each processor system having a processor executing a load locked and store conditional instructions,

said processor executing the load locked instruction to read a memory block, and said processor in response to executing the load locked instruction issuing a read modify system command to read said block and to take ownership of said block by said processor, and also setting a lock flag for the address of said memory block, and writing a value of said memory block into a cache of said processor as a cache copy of said memory block;

a MAF state machine resetting said lock flag upon receipt of an invalidate message for said cache copy of said memory block, if any said invalidate messages are received by said processor;

a controller waiting for a selected time interval before said processor surrenders ownership of said memory block upon receipt of an ownership request message, if any is received by said processor after execution of said load locked instruction;

said processor executing the store conditional instruction, and said processor testing said lock flag, and if said lock flag is set, writing to said cache copy of said memory block.

23. A computer readable media comprising: computer readable instructions for the practice of the method of claim 1 written thereon.

24. Electromagnetic signals propagating on a computer network, said electromagnetic signals carrying information for the practice of the method of claim 1.